

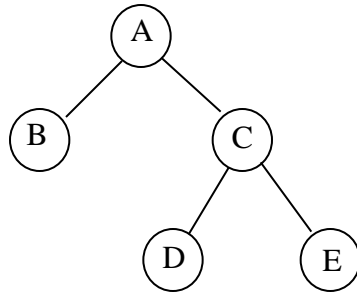
**Multiple choice test 8 (kapitel 15, 16 og 17)**

1. Hvad er den værste omkostning for en enkelt `push`-operation, hvis der benyttes array-fordobling?
  - a.  $O(1)$
  - b.  $O(\log N)$
  - c.  $O(N)$
  - d.  $O(N \log N)$
  - e. ingen af ovennævnte muligheder
2. Hvad er den værste totale omkostning for  $N$  konsekutive `stack`-operationer, hvis der benyttes array-fordobling?
  - a.  $O(1)$
  - b.  $O(\log N)$
  - c.  $O(N)$
  - d.  $O(N \log N)$
  - e. ingen af ovennævnte muligheder
3. Hvad sker der, når "wraparound" benyttes ved implementeringen af en kø?
  - a. Hvis `front` går ud over den sidste array-position, bliver dens værdi sat tilbage til den første position i arrayet.
  - b. Hvis `back` går ud over den sidste array-position, bliver dens værdi sat tilbage til den første position i arrayet.
  - c. Såvel (a) som (b).
  - d. Hverken (a) eller (b).
4. Hvad er størrelsen af køen, hvis `front == back`?
  - a. 0
  - b. 1
  - c. 2
  - d. et svar kan ikke gives
  - e. ingen af ovennævnte muligheder
5. Hvor foretages `push` og `pop` i en hæftet liste implementation af en stak?
  - a. `push` foran det første element, `pop` det første element
  - b. `push` efter det sidste element, `pop` det sidste element
  - c. `push` efter det sidste element, `pop` det første element
  - d. `push` foran det første element, `pop` det sidste element
  - e. `push` efter det første element, `pop` det første element
6. Hvor foretages `enqueue` og `dequeue` i en hæftet liste implementation af en kø?
  - a. `enqueue` foran det første element, `dequeue` det første element
  - b. `enqueue` efter det sidste element, `dequeue` det sidste element
  - c. `enqueue` efter det sidste element, `dequeue` det første element
  - d. `enqueue` foran det første element, `dequeue` det sidste element
  - e. `enqueue` efter det første element, `dequeue` det første element

7. Hvilken af følgende operationer er understøttet ikke effektivt af en enkelt-hægtet liste?
  - a. tilgang til element på den aktuelle position
  - b. indsættelse efter den aktuelle position
  - c. flytning til positionen umiddelbart foran den aktuelle position
  - d. flytning til positionen umiddelbart efter den aktuelle position
  - e. ingen af ovennævnte muligheder
8. Hvilken sætning indsætter et element  $x$  efter positionen  $current$ ?
  - a. `current = new ListNode(x, current);`
  - b. `current = new ListNode(x, current.next);`
  - c. `current.next = new ListNode(x, current);`
  - d. `current.next = new ListNode(x, current.next);`
  - e. ingen af ovennævnte muligheder
9. Antag at der anvendes et listehoved. Hvilket af følgende udsagn indikerer da, at en liste indeholder præcis 1 element?
  - a. `header == null`
  - b. `header != null`
  - c. `header != null && header.next != null`
  - d. `header != null && header.next != null && header.next.next == null`
  - e. ingen af ovennævnte muligheder
10. Hvilken betingelse er true, hvis og kun hvis en dobbelt-hægtet liste med `head`- og `tail`-knuder er tom?
  - a. `head == null`
  - b. `tail == null`
  - c. `head == tail`
  - d. `head.next == tail.prev`
  - e. ingen af ovennævnte muligheder
11. Hvor mange ændringer af `next` og `prev` kræves for at indsætte et element i en dobbelt-hægtet liste?
  - a. ingen
  - b. 1 af `next`, 1 af `prev`
  - c. 2 af `next`, 2 af `prev`
  - d. 3 af `next`, 3 af `prev`
  - e. ingen af ovennævnte muligheder
12. Hvilken af følgende traversingsmetoder kræver mere end lineært tid i værste tilfælde?
  - a. inorder
  - b. level order
  - c. postorder
  - d. preorder
  - e. ingen af ovennævnte muligheder
13. I hvilken af følgende travewrseringmetoder behandles en knude efter, at de rekursive kald af børnene er udført?
  - a. inorder
  - b. level order
  - c. postorder
  - d. preorder
  - e. ingen af de ovennævnte muligheder

14. Hvad er det minimale antal knuder i et binært træ med  $L$  blade?
- $\log L$
  - $2L - 1$
  - $2L$
  - $2^L$
  - ingen af ovennævnte muligheder
15. Lad  $c(t)$  være antallet af blade i et binært træ med rod  $t$ . Antag at  $isLeaf(t)$  returnerer 1, hvis  $t$  er et blad; ellers 0. Hvilket af følgende ligninger kan føre til en rekursiv implementering af  $c(t)$ .
- $c(t) = c(t.left) + c(t.right)$
  - $c(t) = c(t.left) + c(t.right) + 1$
  - $c(t) = c(t.left) + c(t.right) + isLeaf(t)$
  - $c(t) = c(t.left) + c(t.right) + isLeaf(t) + 1$
  - ingen af ovennævnte muligheder
16. Hvor mange gange placeres en knude på stakken ved en inorder-traversering?
- 0
  - 1
  - 2
  - 3
  - ingen af ovennævnte muligheder
17. Hvilke traverseringer benytter ikke en stak?
- inorder
  - level order
  - postorder
  - preorder
  - alle 4 ovennævnte

De næste 3 spørgsmål omhandler nedenstående træ:



18. Hvilken af følgende traverseringer giver rækkefølgen ABCDE?
- inorder
  - level order
  - postorder
  - preorder
  - to af de ovennævnte muligheder
19. Hvilken af følgende rækkefølger fremkommer ved inorder-traversering?
- ABCDE
  - ABDCE
  - BDECA
  - EDCBA
  - ingen af ovennævnte muligheder
20. Træets højde er
- 0
  - 1
  - 2
  - 3
  - to af de ovennævnte muligheder